Like Bayesian networks, possibilistic ones compactly encode joint uncertainty representations over a set of variables. Learning possibilistic networks from data in general and from imperfect or scarce data in particular, has not received enough attention. Indeed, only few works deal with learning the structure and the parameters of a possibilistic network from a dataset. This paper provides a preliminary comparative empirical evaluation of two approaches for learning the parameters of a possibilistic network from empirical data. The first method is a possibilistic approach while the second one first learns imprecise probability measures then transforms them into possibility distributions by means of probability-possibility transformations. The comparative evaluation focuses on learning belief networks on datasets with missing data and scarce datasets.
1. Introduction

Graphical belief models are compact and powerful representations of uncertain information. They allow to factorize a joint uncertainty measure such as a joint probability measure over a set of variables into a set of local measures. This factorization is often based on the conditional independence relations. Belief networks are either built from information elicited directly from experts or learnt automatically by machine learning techniques from empirical data. Possibilistic formalisms are more suitable for representing qualitative and incomplete information. However, there are only few works dealing with learning possibilistic networks from data (Kruse, Borgelt, 1995; Borgelt, Kruse, 1998; Haddad et al., 2015b; 2015a). In particular, learning a possibilistic network may be sound in case of small datasets or datasets with missing or imprecise information (Dubois, Prade, 2016; Haddad et al., 2015a; 2015b).

Learning a graphical belief model comes down in general to i) learn the graphical component, also called structure (namely, extract and encode the independence relationships) and ii) learn the parameters (fill the local tables) associated with each variable. In this paper, we focus on parameter learning of possibilistic networks. Namely, given a structure and a dataset, the goal is to assess local possibility tables of each variable in the context of its parents. The main contributions of the paper are:

- An empirical comparison of two main approaches for learning possibility distributions from data on synthetic datasets. This evaluation compares the networks learnt using two different approaches using a generalized form of the information affinity measure (Jenhani et al., 2007).
- An empirical comparison of learning naive possibilistic network classifiers from real datasets. The evaluation here aims to compare the predictive power of possibilistic classifiers learnt from small datasets containing missing data.

2. Graphical belief models

Belief graphical models such as Bayesian (Darwiche, 2009), credal (Cozman, 2000) and possibilistic networks (C. Borgelt, Kruse, 2003) are powerful tools for representing and reasoning with uncertain information. Bayesian networks allow to compactly encode a probability distribution thanks to the conditional independence relationships existing between the variables. Credal networks, based on the theory of credal sets, generalize Bayesian networks in order to allow some flexibility regarding the model parameters. They are for instance used in robustness analysis and for encoding incomplete and ill-known knowledge and reasoning with the knowledge of groups of experts. Possibilistic networks are the counterparts of Bayesian networks based on possibility theory (Dubois, Prade, 1988; Zadeh, 1999), more suited for handling imperfect, qualitative and partial information.

2.1. Bayesian networks

Bayesian networks \( (\mathcal{BN}) \) are well-known probabilistic graphical models
Possibilistic networks parameter learning (Darwiche, 2009). They allow to compactly represent a probability distribution over a set of variables of interest. A BN = <G, Θ> is specified by:

- A graphical component G with nodes and arcs forming a directed acyclic graph (DAG). Each node represents a variable \( A_i \) of the modeled problem and the edges encode independence relationships among variables.

- A quantitative component Θ, where each variable \( A_i \) is associated with a local probability distribution \( p(A_i | \text{par}(A_i)) \) for \( A_i \) in the context of its parents \( \text{par}(A_i) \).

The joint probability distribution encoded by a Bayesian network is computed using the following chain rule:

\[
P(A_1, \ldots, A_n) = \prod_{i=1}^{n} P(A_i | \text{par}(A_i))
\]  

2.2. Credal networks

Credal networks are probabilistic graphical models based on imprecise probabilities. These latter could be encoded using different representations such as the vertex-based representation (where the credal set is convex and it is characterized by its extreme points), the interval-based representation (where each event is associated with an interval of probability degrees), etc. A credal network \( CN = <G, K> \) is a probabilistic graphical model where

- \( G = <V, E> \) is a directed acyclic graph (DAG) encoding conditional independence relationships where \( V = \{A_1, A_2, \ldots, A_n\} \) is the set of variables of interest (\( D_i \) denotes the domain of variable \( A_i \)) and \( E \) is the set of arcs of \( G \).

- \( K = \{K_1, K_2, \ldots, K_n\} \) is a collection of local credal sets, each \( K_i \) is associated with the variable \( A_i \) in the context of its parents \( \text{par}(A_i) \).

Such credal networks are called separately specified credal networks as the only constraints on probabilities are specified in local tables for each variable in the context of its parents. Note that in practice, in local tables, one can either specify a set of extreme points characterizing the credal set as in JavaBayes\(^1\) software or directly local interval-based probability measures.

A credal network \( CN \) can be seen as a set of Bayesian networks \( BN \)’s, each encoding a joint probability distribution. One can compute an interval-based joint probability distribution as follows:

\[
\underline{P}(a_1a_2\ldots a_n) = \min_{p \in \text{ext}(CN)} (p(a_1a_2\ldots a_n))
\]

\[
\overline{P}(a_1a_2\ldots a_n) = \max_{p \in \text{ext}(CN)} (p(a_1a_2\ldots a_n))
\]

In Equations 2 and 3, \( p(a_1a_2\ldots a_n) \) is computed with the chain rule (see Equation 1). Note that the vertices of \( \text{ext}(CN) \) can be obtained by considering only the set of vertices of the local credal sets \( K_i \) associated with the variables (Cozman, 2000).

\(^1\) http://www.cs.cmu.edu/~javabayes/Home/
2.3. Possibilistic networks

A possibilistic network $\mathcal{P}N=\langle G, \Theta \rangle$ is specified by:

i) A graphical component $G$ consisting of a directed acyclic graph (DAG) where vertices represent the variables and edges represent direct dependence relationships between variables.

ii) A numerical component $\Theta$ allowing to weight the uncertainty relative to each variable using local possibility tables. The possibilistic component consists in a set of local possibility tables $\theta_i = \pi(A_i|\text{par}(A_i))$ for each variable $A_i$ in the context of its parents $\text{par}(A_i)$ in the network $\mathcal{P}N$.

Note that all the local possibility distributions $\theta_i$ must be normalized, namely $\forall i=1..n$, for each parent context $\text{par}(a_i)$, $\max_{a_i \in D_i}(\pi(a_i | \text{par}(a_i))=1$. See in Figure 1 an example of a Bayesian network and an example of a possibilistic one.

In the possibilistic setting, the joint possibility distribution is factorized using the following possibilistic counterpart of the chain rule:

$$\pi(a_1, a_2, ..., a_n) = \otimes_{i=1}^n(\pi(a_i | \text{par}(a_i))).$$  \hspace{1cm} (4)

where $\otimes$ denotes the product or the min-based operator depending on the quantitative or the qualitative interpretation of the possibilistic scale (Dubois, Prade, 1988). In this work, we are interested only in product-based possibilistic networks since we view possibility degrees as upper bounds of probability degrees.

3. Learning the parameters of a possibilistic network

Learning the parameters of a possibilistic network is the problem of assessing the entries of local possibility tables $\pi(A_i|\text{par}(A_i))$ for each variable $A_i$ given a structure $S$ and a dataset $D$. The structure here is assumed to be given (eg. when learning naive classifiers, the structure is fixed in advance by assumption) or learnt automatically. There are basically two ways to learn the parameters (Haddad et al., 2015a ; 2015b ; Benferhat et al., 2015 ; Slimen et al., 2013). i) Transformation-based approach ($TA$ for short) and ii) Possibilistic-based approach ($PA$ for short). Note that the authors in (Serrurier, Prade, 2015) propose a possibilistic-based method for learning the structure of a Bayesian network.

3.1. Transformation-based approach

This approach consists in first learning the parameters of a probabilistic network then transforming the obtained probabilistic network into a possibilistic one (Haddad et al., 2015a ; 2015b ; Benferhat et al., 2015 ; Slimen et al., 2013). Transformations can be useful in various contexts such as using the existing tools (e.g. algorithms and software) developed in one setting rather than developing everything from scratch for the other setting or exploiting information provided in different uncertainty languages as it is often the case in some multiple expert applications.
Many probability-possibility transformations exist (Dubois et al., 2004; Klar, Geer, 1993; Zadeh, 1999) but most of the works deal with the desirable properties and propose transformations satisfying such properties. Among these transformations, the optimal transformation (OT) (Dubois et al., 2004) is defined as follows:

$$\pi_i = \sum_{j \mid p_j \leq p_i} p_j,$$

where $\pi_i$ (resp. $p_i$) denotes $\pi(\omega_i)$ (resp. $p(\omega_i)$). The transformation of Equation 5 transforms $p$ into $\pi$ and guarantees that the obtained possibility distribution $\pi$ is the most specific (hence most informative) one that is consistent and preserving the order of interpretations.

In case where the probabilistic model is a credal one (Bayesian networks where local tables encode sets of probability measures), one can make use of imprecise probability-possibility transformations turning for instance an interval-based probability distribution $p$ into $\pi$ and guarantees that the obtained possibility distribution $\pi$ is the most specific (hence most informative) one that is consistent and preserving the order of interpretations.

In (Destercke et al., 2007), the authors claim that any upper generalized $R$-cumulative distribution $F$ built from one linear extension can be viewed as a possibility distribution and it also dominates all the probability distributions that are compatible with the IPD. Let $C_i$ be a linear extension compatible with the partial order $M$ induced by an IPD. Let $\phi_1, \phi_2, \ldots, \phi_n$ be subsets of $\Omega$ such that $\phi_i = \{\omega_j \mid \omega_j \leq C_i \omega_i\}$. The upper cumulative distribution $F$ built from one linear extension $C_i$ is as follows (see (Desierecke et al., 2007) for more details):

$$F(\phi_i) = \min(\sum_{\omega_j \in \phi_i} u_j, 1 - \sum_{\omega_j \notin \phi_i} l_j)$$

The obtained cumulative distribution $F$ is a possibility distribution dominating the IPD and it is such that $F(\phi_i) = \Pi(A_i)$. The advantage of such a transformation, also called p-box transformation, is its low computational cost (linear in the size of domains) and the fact that the obtained distribution is better in terms of specificity (meaning that the transformation process losses less information).

---

2. Let $\pi'$ and $\pi''$ be two possibility distributions, $\pi'$ is more specific than $\pi''$ iff $\forall \omega_i \in \Omega, \pi'(\omega_i) \leq \pi''(\omega_i)$.
Now, using a probability-possibility transformation, one can turn a Bayesian or credal network $\mathcal{BN}$ into a possibilistic network $\mathcal{PN}$ only by turning every local probability table into a possibilistic one while preserving the structure of the network in order to preserve the conditional independences encoded by $\mathcal{BN}$.

**Example 1.** — Let $\mathcal{BN}$ be the Bayesian network of Figure 1 over two variables $A$ and $B$ having the domains $D_A=\{a_1, a_2\}$ and $D_B=\{b_1, b_2, b_3\}$ respectively.

![Figure 1. Example of a Bayesian network $\mathcal{BN}$ and the possibilistic network $\mathcal{PN}$ obtained from $\mathcal{BN}$ using the optimal transformation $OT$.](image)

3.2. Possibilistic-based approach

One view of possibility theory is to consider a possibility distribution $\pi$ on a variable $A_i$ as a *counter function* of a random set (Shafer et al., 1976) pertaining to $D_i$, the domain of $A_i$. A random set in $D_i$ is a random variable which takes its values on subsets of $D_i$. More formally, let $D_i$ be a finite domain. A basic probability assignment or mass function is a mapping $m : 2^{D_i} \rightarrow [0, 1]$ such that $\sum_{a_i \subseteq D_i} m(a_i) = 1$ and $m(\emptyset) = 0$. A set $a_i \subseteq D_i$ such that $m(a_i) > 0$ is called a focal set.

The possibility degree of an event $a_i$ is the probability of the possibility of the event i.e. the probability of the disjunction of all events (focal sets) $a_i'$ in which this event is included (C. Borgelt et al., 2009):

$$\pi(a_i) = \sum_{a_i' \mid a_i \cap a_i' \neq \emptyset} m(a_i')$$  \hspace{1cm} (7)

A random set is said to be *consistent* if there is at least one element $a_i$ contained in all focal sets $a_i'$ and the possibility distribution induced by a consistent random set is, thereby, normalized. Exploring this link between possibility theory and random sets theory has been extensively studied, in particular, in learning tasks, we cite for instance (C. Borgelt et al., 2009; Joslyn, 1997). In what follows, we investigate this link to learn possibilistic networks parameters from imprecise data.

Given a DAG and an imprecision degree $S_i$, let $D_{ij} = \{d_{ij}^{(l)}\}$ be a dataset relative to a variable $A_i$, $d_{ij}^{(l)} \in D_{ij}$ (resp. $d_{ij}^{(l)} \subseteq D_{ij}$) if data are precise (resp. imprecise). The number of occurrences $A_i = a_{ik}$ such that $Pa(A_i) = j$, denoted
by $N_{ijk}$, is the number of times $A_i = a_{ik}$ such that $Pa(A_i) = j$ appears in $D_{ij}$: 

$$N_{ijk} = \text{card}\{l \text{ s.t. } A_i = a_{ik} \text{ s.t. } Pa(A_i) = j \in d_{ij}\}.$$ 

$$\hat{\pi}(A = a_{ik} | Pa(A_i) = j) = \frac{N_{ijk}}{\sum_{k=1}^{r_i} N_{ijk}} * S_i \quad (8)$$

where $q_i = \text{card}(Pa(X_i))$, $r_i = \text{card}(D_i)$ and $S_i$ corresponds to the imprecision degree relative to a variable $A_i$. To obtain normalized possibility distributions, we divide each obtained distribution by its maximum. It is evident that this operation eliminates $S_i$. However, we could assign to each value of $X_i$ an imprecision degree which could be either set by an expert or inferred from the dataset to learn from.

4. Experimental study 1: How similar are possibilistic networks where the parameters are learnt using different approaches?

This section provides experimental results comparing the two approaches for learning the parameters of a possibilistic network from imprecise data.

4.1. Experimentation setup

In this experiment, given a dataset $D$ and a network structure (DAG) $S$, we compare learning a possibilistic network parameters using two approaches, $TA$ and $PA$. We denote by $G^{TA}$ (resp. $G^{PA}$) the possibilistic network having the structure $S$ and its parameters are learnt over the dataset $D$ using the transformation-based approach $TA$ based on the p-box transformation (resp. the possibilistic-based approach $PA$).

When learning belief graphical models like Bayesian or possibilistic networks from data, the evaluation is generally carried out by comparing reference networks with the learnt ones. Reference networks are graphical models that are either chosen by an expert or randomly generated. From the reference model, a dataset is generated following the distribution encoded by the reference model. This dataset is then used to learn models using the approach to be evaluated. The problem then comes down to compare the learnt model with the reference one. A comparison may take into account only the joint measures encoded by the learnt and the reference models. In addition, one may want also to take into account the structure of the learnt and reference models. Given that we are only interested in comparing possibilistic networks with same structure, there is no need to consider the graphical component in our comparisons.

One simple but costly way of comparing the reference network with the learnt one is to compare only the joint distribution encoded by the reference model with the learnt model distribution. An example of similarity measure for possibility distributions is information affinity (Jenhani et al., 2007). However the size of the distribution may be very huge (it fact, it is exponential in the number of variables of the network) making it impossible to compare joint possibility distributions. We propose a heuristic method that compares the networks’ local distributions locally and aggregates the results to
provide an overall similarity score of two possibilistic networks. Information affinity (Jenhani et al., 2007) is defined as follows:

\[ InfoAff \left( \pi_1, \pi_2 \right) = 1 - \frac{d(\pi_1, \pi_2) + Inc(\pi_1, \pi_2)}{2} \] (9)

where \( d(\pi_1, \pi_2) \) represents the mean Manhattan distance between possibility distributions \( \pi_1 \) and \( \pi_2 \) and defined as follows: \( d(\pi_1, \pi_2) = \frac{1}{N} \sum_{i=1}^{N} |\pi_1(\omega_i) - \pi_2(\omega_i)| \).

As for \( Inc(\pi_1, \pi_2) \), it is a measure of inconsistency and it assesses the conflict degree between \( \pi_1 \) and \( \pi_2 \). Namely, \( Inc(\pi_1, \pi_2) = \max_{\omega \in \Omega} (\pi_1(\omega) \land \pi_2(\omega)) \) where \( \pi_1(\omega) \land \pi_2(\omega) \) denotes a combination operation of two possibility distributions. In (Jenhani et al., 2007), the \( \min \) operator is used in a qualitative setting. In a quantitative setting, a product operator can be used as well.

**Example 2.** Let \( \Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\} \) and let \( \pi_1 \) and \( \pi_2 \) be two possibility distributions such that \( \pi_1 = (1, 8, 4, 0) \) and \( \pi_2 = (9, 1, 2, 0) \). Then \( InfoAff(\pi_1, \pi_2) = 1 - \frac{d(\pi_1, \pi_2) + Inc(\pi_1, \pi_2)}{2} = 1 - \frac{d(0.175, 0.1)}{2} = 0.8625 \).

The measure of Equation 9 satisfies many natural properties such as non-negativity, symmetry, upper bound and non-degeneracy, lower bound, inclusion and permutation. However, it cannot be directly applied for assessing the similarity of two possibilistic networks. Now, in order to assess the similarity between two possibilistic networks \( G_1 \) and \( G_2 \) having the same structure (same DAG), then it may be relevant to compare locally every local possibility distribution \( \pi_{i} \) in the network \( G_1 \) with \( \pi_{2} \), namely its corresponding distribution in \( G_2 \). This can be done for instance using an aggregation function (in this work, we use the mean) that takes into account all the local distributions and returns a global similarity score between \( G_1 \) and \( G_2 \).

\[ GrInfoAff(G_1, G_2) = \frac{1}{N} \left( \sum_{i=1}^{N} InfoAff(\pi_{i}^{1}, \pi_{i}^{2}) \right) \] (10)

where \( N \) stands for the number of local distributions of the networks.

### 4.2. Benchmarks

We first generated a set of possibilistic networks with different features (number of variables, number of parents per variable, rate of imprecise data, etc.). For each possibilistic network \( G \), we generate datasets according to \( G \). More precisely, for each possibilistic network \( G \) (characterized by its number of variables denoted \( \# \text{variables} \), the mean number of parents per node denoted \( \mu \text{variables} \) and the mean domain size of variables \( \mu \text{domain} \)), we generate many datasets (with different sizes). Regarding the dataset generation process, it consists in generating an imprecise dataset representative of its possibility distribution. The sampling process constructs a database of \( N \) (predefined) observations by instantiating all variables w.r.t. their possibility distributions using the \( \alpha \)-cut notion expressed as follows:

\[ \alpha - \text{cut}_{A_{i}} = \{ a_{i} \in D_{i} \text{ s.t. } \pi(a_{i}) \geq \alpha \} \] (11)
where $\alpha$ is randomly generated from $[0,1]$.

Obviously, variables are most easily processed w.r.t. a topological order, since this ensures that all parents are instantiated. Instantiating a parentless variable corresponds to computing its $\alpha$-cut. Instantiating a conditioned variable $A_i$ s.t. $Pa(A_i) = A$ corresponds to computing the $\alpha$-cut of $\pi(A_i|Pa(A_i) = A)$ computed as follows:

$$\pi(A_i|Pa(A_i) = A) = \max_{a_i \in A} (\pi(A_i|a_i), \pi(a_i))$$ (12)

Table 1 gives the details on the generated possibilistic networks and the corresponding datasets.

**Table 1. Datasets properties used in experiments 1.**

<table>
<thead>
<tr>
<th>Name</th>
<th># variables</th>
<th>$\mu$ parents</th>
<th>$\mu$ domain</th>
<th># datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net10</td>
<td>10</td>
<td>1.6</td>
<td>3.9</td>
<td>9</td>
</tr>
<tr>
<td>Net20</td>
<td>20</td>
<td>2.65</td>
<td>3.41</td>
<td>8</td>
</tr>
<tr>
<td>Net30</td>
<td>30</td>
<td>2.76</td>
<td>3.48</td>
<td>7</td>
</tr>
</tbody>
</table>

### 4.3. Results

Table 2 gives the results of computing the similarity on each dataset $D_i$, the possibilistic network $G_{TA}^i$ (resp. $G_{PA}^i$) learnt using the $TA$ (resp. $PA$) approach with the reference network $G_i$ used to generate $D_i$.

**Table 2. Results of experiments 1.**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$TA$</th>
<th>$PA$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net10</td>
<td>0.63</td>
<td>0.8</td>
</tr>
<tr>
<td>Net20</td>
<td>0.64</td>
<td>0.8</td>
</tr>
<tr>
<td>Net30</td>
<td>0.67</td>
<td>0.8</td>
</tr>
</tbody>
</table>

The results of Table 2 show that on the one hand the learnt possibilistic networks using the $TA$ approach are close to the reference ones. Namely, they have rather a good similarity with the reference possibilistic networks used to generate the datasets. Moreover, the obtained similarity scores do not seem to be affected by the number of variables, variable domains size, etc. Regarding the possibilistic networks learnt using the $PA$ approach, their similarity scores are slightly better, but this is expected as the datasets generation process and the $PA$ approach have the same view of possibility degrees. Such results also rise the issue of similarity measures on possibilistic networks which is still an open issue.

### 5. Experimental study 2: Predictive power of possibilistic network classifiers

where the parameters are learnt using different approaches

In this section, we evaluate the predictive power of credal network classifiers (Corani, Zaffalon, 2008) with respect to their possibilistic network counterparts. More
precisely, we compare on many datasets the classification efficiency of naive credal classifier \((NCC)\) for short) and the corresponding possibilistic classifiers obtained either using the possibilistic-based approach \((PNC_{PA})\) or using the transformation-based approach \((PNC_{TA})\). Moreover, we compare our results to naive Bayes classifier \((NBC)\) as a baseline.

5.1. Experimentation setup

In order to evaluate the \(NCC\) classifier, we use the following measures used in (Corani, Zaffalon, 2008).

- **Determinacy (Det)**: It is the percentage of predictions outputting a unique (precise) class label.
- **Single-Accuracy (SiAcc)**: It denotes the percentage of correct classifications when the predictions of \(NCC\) are precise.
- **Set-Accuracy (SetAcc)**: It is the proportion of imprecise predictions containing the right class label.

The evaluation mode used in this experiment is a 10-fold cross validation.

5.1.1. Benchmarks

The experimental study is carried out on the following datasets where some data values are missing. Note the missing data is assumed to be not missing at random. The first four datasets of Table 3 are real datasets used in the literature for evaluating classifiers with missing data (http://sci2s.ugr.es/keel/missing.php). The remaining ones are collected from different sources.

<table>
<thead>
<tr>
<th>Name</th>
<th># instances</th>
<th># variables</th>
<th># classes</th>
<th>% missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>breast</td>
<td>286</td>
<td>9</td>
<td>2</td>
<td>4%</td>
</tr>
<tr>
<td>housevotes</td>
<td>435</td>
<td>16</td>
<td>2</td>
<td>24%</td>
</tr>
<tr>
<td>mushroom</td>
<td>8124</td>
<td>22</td>
<td>2</td>
<td>31%</td>
</tr>
<tr>
<td>post-operative</td>
<td>90</td>
<td>8</td>
<td>3</td>
<td>3%</td>
</tr>
<tr>
<td>audiology</td>
<td>226</td>
<td>70</td>
<td>24</td>
<td>98%</td>
</tr>
<tr>
<td>sick</td>
<td>3772</td>
<td>30</td>
<td>2</td>
<td>20%</td>
</tr>
<tr>
<td>primary-tumor</td>
<td>339</td>
<td>18</td>
<td>21</td>
<td>46%</td>
</tr>
<tr>
<td>kr-vs-kp</td>
<td>3196</td>
<td>37</td>
<td>2</td>
<td>0 %</td>
</tr>
<tr>
<td>soybean</td>
<td>683</td>
<td>36</td>
<td>19</td>
<td>18%</td>
</tr>
<tr>
<td>crx</td>
<td>690</td>
<td>16</td>
<td>2</td>
<td>2%</td>
</tr>
</tbody>
</table>

5.1.2. Results

Table 4 gives the results of evaluating the \(NCC\) classifier on the datasets of Table 3.
Tableau 4. Results of the NCC classifier on the datasets of Table 3.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Det</th>
<th>SiAcc</th>
<th>SetAcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>breast</td>
<td>92.43 %</td>
<td>74.08 %</td>
<td>100 %</td>
</tr>
<tr>
<td>housevotes</td>
<td>99.52 %</td>
<td>90.26 %</td>
<td>100 %</td>
</tr>
<tr>
<td>mushroom</td>
<td>96.10 %</td>
<td>99.56 %</td>
<td>100 %</td>
</tr>
<tr>
<td>post-operative</td>
<td>49.67 %</td>
<td>67.57 %</td>
<td>84.36 %</td>
</tr>
<tr>
<td>audiology</td>
<td>7.76 %</td>
<td>99.55 %</td>
<td>99.03 %</td>
</tr>
<tr>
<td>sick</td>
<td>98.93 %</td>
<td>97.54 %</td>
<td>100 %</td>
</tr>
<tr>
<td>primary-tumor</td>
<td>13.59 %</td>
<td>77.11 %</td>
<td>63.37 %</td>
</tr>
<tr>
<td>kr-vs-kp</td>
<td>99.18 %</td>
<td>88.16 %</td>
<td>100 %</td>
</tr>
<tr>
<td>soybean</td>
<td>47.38 %</td>
<td>92.56 %</td>
<td>97.85 %</td>
</tr>
<tr>
<td>crx</td>
<td>94.01 %</td>
<td>86.34 %</td>
<td>100 %</td>
</tr>
</tbody>
</table>

The results of Table 4 show good single accuracy rates with high determinacy rates except for the post-operative, audiology, primary-tumor and soybean datasets. Typically, it’s on small datasets with many classes where the NCC is not efficient.

Table 5 gives the results of evaluating the NBC (Naive Bayes Classifier), PNC\textsubscript{TA} and PNC\textsubscript{PA} classifiers on the datasets of Table 3.

Tableau 5. Results of the NBC, PNC\textsubscript{PA} and PNC\textsubscript{TA} classifiers on the datasets of Table 3.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>% of correct classifications</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NBC</td>
</tr>
<tr>
<td>breast</td>
<td>72.88%</td>
</tr>
<tr>
<td>housevotes</td>
<td>90.11 %</td>
</tr>
<tr>
<td>mushroom</td>
<td>95.73 %</td>
</tr>
<tr>
<td>post-operative</td>
<td>68.11 %</td>
</tr>
<tr>
<td>audiology</td>
<td>72.79%</td>
</tr>
<tr>
<td>sick</td>
<td>96.97%</td>
</tr>
<tr>
<td>primary-tumor</td>
<td>49.54%</td>
</tr>
<tr>
<td>kr-vs-kp</td>
<td>87.82%</td>
</tr>
<tr>
<td>soybean</td>
<td>92.66%</td>
</tr>
<tr>
<td>crx</td>
<td>85.38%</td>
</tr>
</tbody>
</table>

Results of Table 5 show that classifiers NBC, PNC\textsubscript{PA} and PNC\textsubscript{TA} have most of the time comparable results in terms of correct classification rates on some datasets but they show real performances on some other datasets. This is also valid for the results of the NCC classifier. Now, comparing PNC\textsubscript{PA} and PNC\textsubscript{TA}, this latter achieves better results on two datasets while the former has better classification rates on the two other datasets. It is not obvious what makes a given approach better, a thorough analysis of the properties of the datasets is needed to help understanding such results.
6. Discussions and concluding remarks

This paper provides preliminary results comparing two methods for assessing the parameters of a possibilistic network given a structure and a dataset. The first experiment shows that the possibilistic-based method learns better and more information in terms of information affinity than the method based on the probability-possibility transformation. This is not surprising since the data was generated according to the possibility distributions of the reference networks. This also confirms that there is some information loss when transforming probability distributions into possibilistic ones. Another important result is that the classifiers based on possibilistic networks have somehow comparable efficiency with naive Bayes and credal classifiers. On the other hand, the possibilistic classifiers where the parameters have been learned with two different approaches have basically comparable results. Overall, these results show that no system really outperforms the others on all the datasets. Such results are preliminary but encouraging, a further comparative study on a large number of benchmarks and problems (classification and inference in general), will be needed to really compare the two approaches. The question that also arises is how to know for a given or a particular dataset which kind of classifier to use?

Acknowledgements

This work is done with the support of a CNRS funded project PEPS FaSciDo 2015 - MAP-POS.

Bibliographie


